Seminar 1

Data Storage Paradigms, IV1351

Ermia Ghaffari
ermia@kth.se

2023-11-07

1 Introduction

In the present era, it is important to manage information and data. For different organizations, managing data has a direct impact on their success and client satisfaction. Managing data will also play a crucial role in the efficiency and security of different projects. This report is about designing a conceptual model for the database of a music school called Soundgood.

1.1 Soundgood music school overview

Soundgood music school is a school that allows students to choose their learning journey in the way they desire. Students can take individual or group lessons based on the level they have. There are also ensembles with different genres where students can play different instruments at the same time. Each lesson has an instructor assigned to it, and the administrative staff can assign a given instructor for a lesson. Furthermore, students can also borrow instruments if available. Since the reader has the same knowledge regarding the project as the author, the Soundgood school is briefly described.

1.2 Requirements

The conceptual model must clearly cover all the project descriptions outlined on Canvas(project descriptions 1.1 and 1.2). In other words, all the data that are related to students, instructors, payments, and instruments should be stored. It is possible to see the main requirements of the solution below.

- The conceptual model should be able to meet all the criteria of Soundgood music school. Thus, the conceptual model designed should be able to keep track of information such as student enrollment, and bookings of the lesson.
- The model should contain the personal information of both instructors and students. For instance, the personal number, name, contact details and the address. The model should also allow the storage of data for a contact person of students.

- There should be time slots for the group lessons and the ensembles. Individual lessons are more flexible and they work like appointments. Hence, CM must contain the lesson schedule.
- The CM should be made in UML or using IE notation. In this case, the IE is used, and the CM contains all the crucial requirements of the school described on Canvas.
- The report should have a discussion that will completely explain the task. Furthermore, an inheritance should be used to achieve a higher grade in the seminar.

This task is done with the collaboration of Daniel Ibrahimi and Esra Salman.

2 Literature Study

Before working with the conceptual model, chapter one, two and three of the book "Data Modeling Using the Entity-Relationship", was read by all the members of the group. The first step is to understand that a conceptual model is like the path between the real world and abstracts of database systems. Furthermore, a conceptual model is a model that a specialist would figure out by talking to the customer and observing what are the requirements of the project. The conceptual model will provide a bigger picture to the people who are working on the design of the database. This is due to the fact that all the technical details are avoided, and more or less the only things shown on the model are the relationships between different entities. In this way, the audience with low technical skills won't be confused when looking at the model(Ramez Elmasri and Navathe, 2016, pp.60–65).

After reading about the theoretical aspects of the conceptual model, it is important to learn about the theoretical aspects of implementing a conceptual model for the Soundgood music school. In this part, the videos provided on Canvas are used in order to be able to design a conceptual model of type ER diagram using IE notation.

3 Method

This part is about the methodology of designing the conceptual method. As mentioned before in the literature study, the group followed the instructions on Canvas. The text editor used in this task is Astah. The activation of the Astah application was performed using the instructions on Canvas.

3.1 Designing the model

The project description on Canvas and the requirements of the Soundgood music school were read carefully. Note that one should read both the Business Overview and Detailed Descriptions on Canvas in order to start with the methodology. The methodology is divided into three different iterations. The first iteration is to find all the common

attributes so that we can use inheritance. The second is to find all the relations and cardinalities of attributes. The third iteration is to think like a user. For instance, how to calculate the salary of an instructor? Each step and detailed description of the methodology is explained in the following paragraphs.

Afterwards, all the nouns that are related and can be used as entities were extracted from the project description. Furthermore, as explained in the requirement part, all the needed school's operations were listed. The next step is to analyse the scope of our database. Additionally, all the relationships and needed entities were discussed in the group.

After identifying the related needed entities, proper names for the entities are chosen. It should be noted that the names of entities were changed during the process due to finding better names. It is also important to note that some entities were also deleted from the model due to the fact that some of the entities were irrelevant to the model Afterwards, proper attributes for the entities were chosen by reading the project description. These attributes were also adjusted during the process of designing the model. Note that the cardinality of different attributes is chosen when creating the attributes of the entities. Furthermore, there are two options when establishing relationships between two different entities. In the videos provided on Canvas, it is explained to draw the relations when creating the entities. However, the group decided to establish the relations afterwards. This is due to the fact that the whole picture of the design would already be established, and putting extra time into figuring out the relationships between entities is avoided.

In the beginning, the group mistook the task as a model where the coding design is presented on the model instead of conceptual mode. However, after a long discussion, the group came to the conclusion that all the detailed information and coding details should be avoided in the conceptual model. This is due to the fact that a person without any technical knowledge should be able to relate to the model designed. Another issue that the group faced was duplicates in the entities. Let's say that a student wants to borrow an instrument. In the beginning, the thoughts were to have the student ID both in the student entity and the rental instrument entity. Then there would be a connection between the student and the rental instrument entity. After discussing with others, the group realised that since the two entities are related and the student entity already contains student ID, we could remove student ID from the rental instrument entity.

In the end, proper verb phrases were chosen for the relationships, and then some notes were added to the main model. Note that the model is implemented one time with inheritance and another time without inheritance. This will be discussed more in the result and discussion part of the essay.

4 Result

Figure 1 below shows how the conceptual model is designed using inheritances. In the discussion part, we will talk more about the differences between using inheritances and not using inheritances. Figure 2 shows the conceptual model for the Soundgood music

school without any inheritances. The figure above shows the relationships between

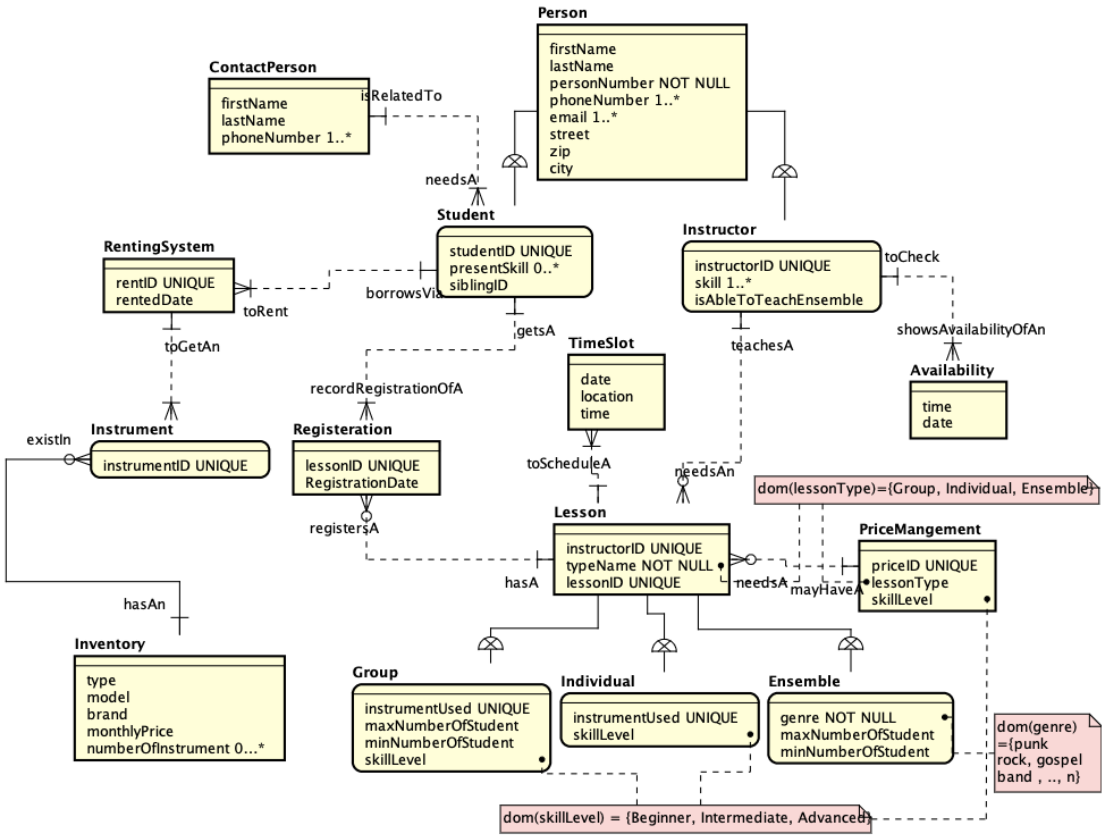

Figure 1: A diagram that represents the conceptual model using inheritance (made by Ermia, Daniel and Esra).

different entities and the attributes that different entities have. At the top of the figure, an entity called Person can be seen. This entity has attributes such as a first name, a last name, a personal number that can not be null, etc. Furthermore, there is an inheritance in the model from the Person entity to the Student entity and the Instructor entity, meaning that whatever attributes the entity Person has, the Student entity and the entity Instructor should have. The entity Student has a student ID, skills and siblingID that might be used later to give discounts. Student is also related to ContactPerson meaning that a student should have a contact person, and the contact person should have first name, last name and a phone number. The entity Student is also connected to the entity RentingSystem and another entity called Registration. In this way, the student can borrow an instrument with a specific ID. At the same time, RentingSystem is connected to Instrument which is connected to inventory which contains all the instruments existing at school.

Additionally, the Student entity is connected to the Registration entity which is later

connected to the Lesson entity. This connection makes the student register for a lesson or multiple lessons. The Lesson entity has an instrument/s ID that is played in that lesson. It also has a type name, and Instructor ID in order to see who is the unique instructor of the course. Group, Individual, and Ensemble entities have the same attributes as the Lesson entity due to the usage of inheritance. However, the Group entity has also the maximum number of students, the minimum number of students, the unique instrument used and the skill level extra as attributes. Individual has instrument used and skill level as attributes and Ensembles have a specific genre, maximum and minimum number of student. The PriceManagement entity is also related to the lessons in order to find the student bills and the instructor's salary.

On the other side, we have the Instructor entity which represents an instructor who is also a person and gets all the attributes of the Person entity in addition to Instructor ID, skill and ability to teach ensembles. Instructor is also connected to the Availability entity which basically is the times that instructors are available.

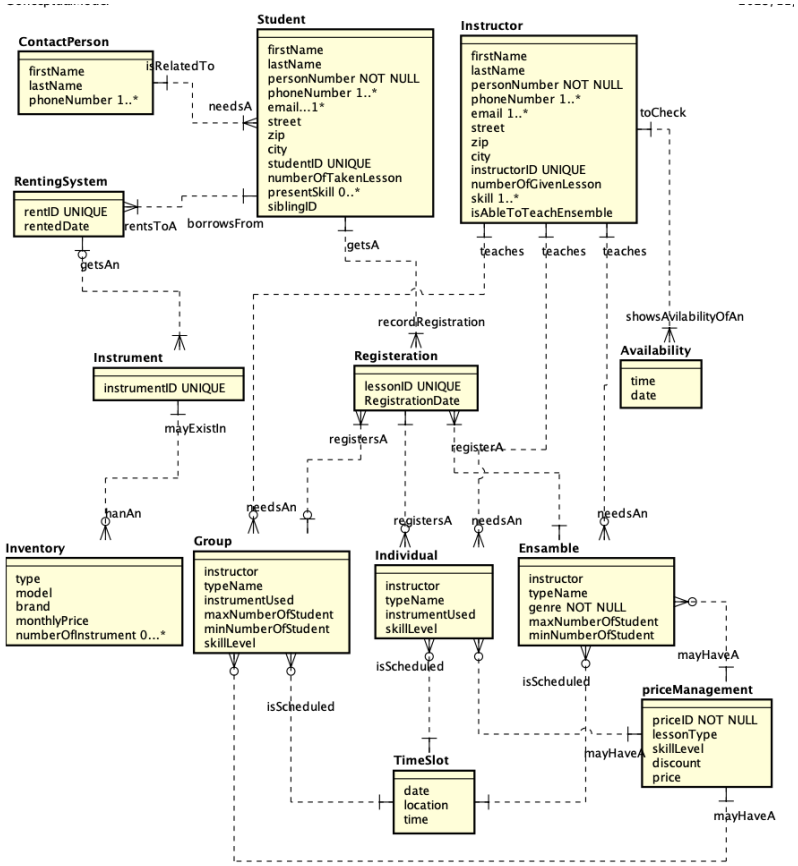

Figure 2: A diagram that represents the conceptual model without using inheritance(made by Ermia, Daniel and Esra).

Figure 2 has almost the same structure as Figure 1. However, Figure 2 represents a conceptual model designed without using any inheritances. All the relations and the entities are the same. However, all the things that a student and an instructor were sharing in person are now duplicated in both entities. The same applies to lessons, groups, individuals and ensembles. There would also be more numbers of relationships between the Group, the Individual, and the Ensemble entities to the rest of the model. This is due to the fact that whatever, the Lesson entity has a relationship to in Figure 1 should have a relationship to the Group, the Individual, and the Ensemble entities. The disadvantage of not using inheritance is explored in the discussion.

5 discussion

As seen in Figure 1, the conceptual model contains all the relevant information of the Soundgood music school. The number of entities is relevant to the needs of Soundgood music school. However, there could be some optimisation on the Registration entity by moving all its attributes to the Student entity. However, getting a relationship between the Student and the Lesson entity directly would be difficult. As can be observed in Figure 1, there are no entities without attributes. Almost all the entities are needed and the only entity that is added extra to make the model more coherent is the Registration entity. Moreover, when designing the conceptual model, proper names for the attributes and entities are used. In addition, conventions are followed in all the names to make the model more coherent for the audience.

The attributes are enough for all the data to be stored. For instance, all the payment data or renting instruments, etc can be stored using the existing attributes in the model. There are also some extra attributes used in the model to make the model more coherent. For instance, by having a glance at the ContactPerson entity for a student, we can see that first name and last name are added. Another example is the email address for the Person entity. The cardinalities of attributes are defined when needed. For instance, a student ID is unique or a personal number can't be null. This can be seen in multiple entities of the model. In some cases, there are notes explaining how things regarding attributes and cardinality work.

In addition, all the important relations are present in the model. There are no significant issues regarding the relations. However, the path from the Student entity to the PaymentManagemnt entity is too long. This would result in an inefficient response time in the designed database. Another important thing to be discussed is the duplicates. Duplicates are avoided in all places except in the Group entity, the Individual entity and the Ensemble entity. The reason for having duplicates is explained in the following paragraph.

An issue that can be pointed out is the attributes that Group and Ensemble entities share with each other or the attributes that Individual and Group entities share with each other. For instance, the maximum and minimum number of students in Group and Ensemble entities are the same. A possibility could be to move these attributes to the Lesson entity. In this case, for the Individual entity, the maximum and minimum number

of students are equal to one. This would cause redundant data. Another solution could be to make another entity and relate that to the Group and Individual entities. However, this would increase the number of entities. If this approach is followed whenever there are some duplicate attributes in different entities, an infinite number of entities might be created in larger databases.

5.1 Using inheritance

As seen in the result section on pages four and five, we can see that using inheritance makes the model more coherent. Using inheritance results in simplifying the structure of the model. It will also centralize the common attributes making the model easy to read. At the same time, not using inheritance can also result in more flexibility. Furthermore, as can be seen, the number of entities is reduced when not using inheritance. However, the number of relations will increase dramatically. This results in a complex model that is not comprehensible to the audience. Furthermore, when modifying and adjusting the model, the usage of inheritance can make the work simpler. This is due to the fact that the shared attributes will be only modified in one entity instead of multiple entities.

6 Link to Github

https://github.com/ermia1230/IV1351.git

7 Reference list

Ramez Elmasri and Navathe, S.B. (2016). Fundamentals of database systems. Hoboken, New Jersey: Pearson, pp.60–65.